

8 盒模型

内容

[8.1 盒尺寸 \(dimensions\)](#)

[8.2 margin、padding和border的示例](#)

[8.3 外边距属性: 'margin-top', 'margin-right', 'margin-bottom', 'margin-left'和'margin'](#)

[8.3.1 合并外边距](#)

[8.4 内边距属性: 'padding-top', 'padding-right', 'padding-bottom', 'padding-left' 和'padding'](#)

[8.5 边框属性](#)

[8.5.1 边框宽度: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width'和'border-width'](#)

[8.5.2 边框颜色: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color'和'border-color'](#)

[8.5.3 边框样式: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style'和'border-style'](#)

[8.5.4 Border shorthand properties: 'border-top', 'border-right', 'border-bottom', 'border-left'和'border'](#)

[8.6 双向环境 \(bidirectional context\) 中行内元素的盒模型](#)

CSS盒模型描述了一个为[文档树](#)中的元素生成的并根据[视觉格式化模型](#)进行布局的矩形框

8.1 盒尺寸 (dimensions)

每个盒都有一块 内容区 (*content area*) (例如, 文本, 一张图片等等) 和周围可选的 *padding*, *border* 和 *margin* 区域, 每块区域的尺寸通过下面定义的属性指定。下图展示了这些区域的联系以及用来引用margin, border和padding块的术语:



margin, border和padding可以分为上, 右, 下和左4部分 (例如, 图中, "LM"表示左外边距, "RP"表示右内边距, "TB"表示上边框等等)

4块区域 (内容, 内边距, 边框和外边距) 中每一块的边界叫一条“边界 (edge)”, 所以每个盒有4条边界:

内容边界或内边界

内容边界环绕着盒的 [width](#) 和 [height](#) 指出的矩形, 通常取决于元素的 [呈现 \(rendered\) 内容](#)。4条内容边界定义了盒的内容框 (*content box*)

内边距边

内边距边界环绕着盒的内边距。如果内边距的宽度为0, 那么内边距边界就和内容边界相同。4条内边距边界定义了盒的内边距框 (*padding box*)

边框边界

边框边界环绕着盒的边框。如果边框的宽度为0, 那么边框边界就和内边距边界相同。4条边

框边界定义了盒的边框框 (*border box*) .
外边距边界或外边界
外边距边界环绕着盒的外边距。如果外边距的宽度为0, 那么外边距边界就和边框边界相同。4条外边距边界定义了盒的外边距框 (*margin box*)

每条边界都可以细分成上, 右, 下和左边界

盒的内容区的尺寸—*content width* 和*content height* —取决于几个因素: 生成该盒的元素是否设置了'width'或'height' 属性, 该盒是否包含文本以及其它盒, 该盒是不是表格等等。盒的宽度和高度在[视觉格式化模型细节](#)章节中讨论

盒的内容, 内边距和边框区域的背景样式由生成 (该盒的) 元素的'background' 属性来指定。外边距背景总是透明的

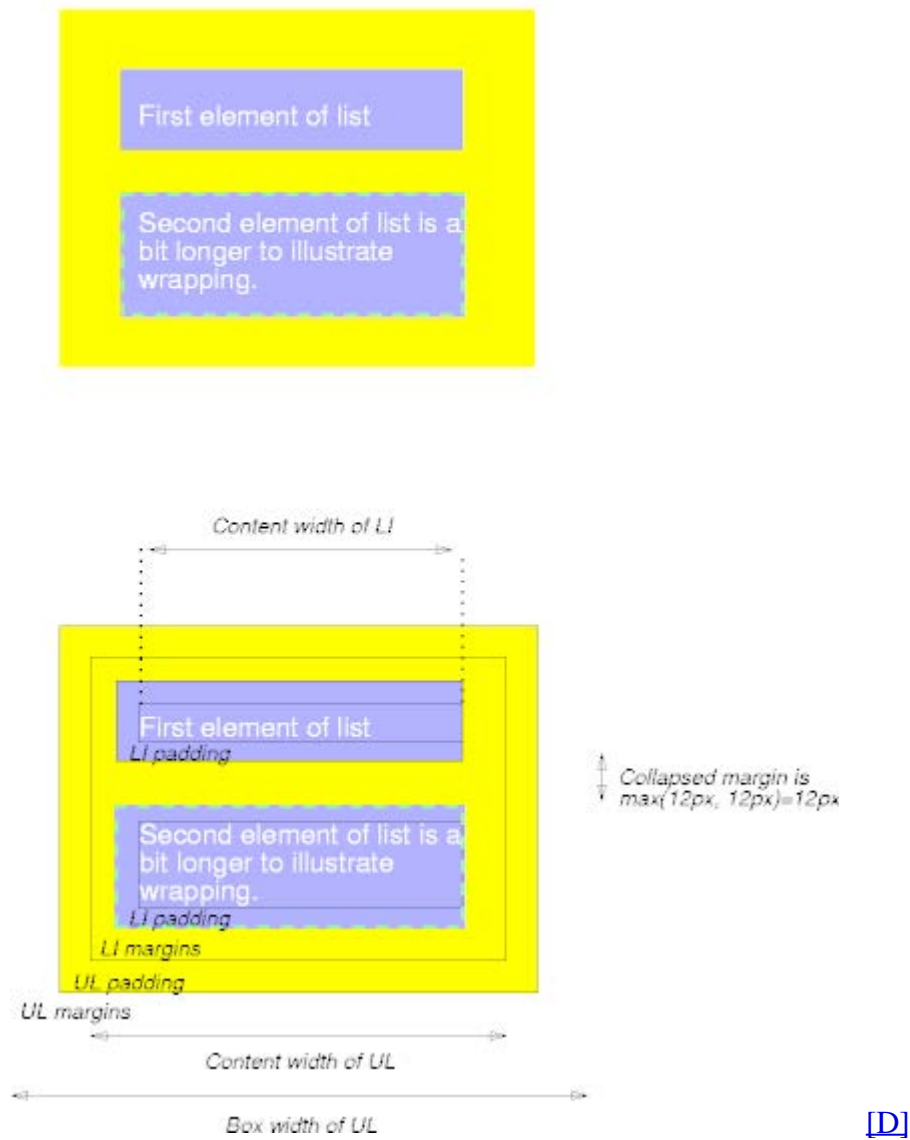
8.2 margin、padding和border的示例

本例描述了margin, 内边距和border之间的相互影响。示例HTML文档:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<HTML>
  <HEAD>
    <TITLE>Examples of margins, padding, and borders</TITLE>
    <STYLE type="text/css">
      UL {
        background: yellow;
        margin: 12px 12px 12px 12px;
        padding: 3px 3px 3px 3px;
      } /* No borders set */
      LI {
        color: white;
        background: blue;
        margin: 12px 12px 12px 12px;
        padding: 12px 0px 12px 12px;
        list-style: none
      } /* text color is white */
        /* Content, padding will be blue */
        /* Note 0px padding right */
        /* no glyphs before a list item */
        /* No borders set */
      LI.withborder {
        border-style: dashed;
        border-width: medium;
        border-color: lime;
      } /* sets border width on all sides */
    </STYLE>
  </HEAD>
  <BODY>
    <UL>
      <LI>First element of list
      <LI class="withborder">Second element of list is
        a bit longer to illustrate wrapping.
    </UL>
  </BODY>
</HTML>
```

产生含有一个拥有2个LI孩子的UL元素的[文档树](#) ((文档树中) 还有其它关系)

下面的第一幅图描述了本例将产生的结果。第二幅图描述了UL元素的margin, 内边距, border和它的子级LI元素的这些属性之间的关系 (图像不是按比例)



注意：

- 每个LI盒的content width 是自顶向下计算的，每个LI盒的包含块是由UL元素建立的
- 每个LI盒的外边距框高度取决于其content height，加上上下的内边距，边框以及外边距。注意LI盒之间的垂直margin合并了
- LI盒的右内边距被设置为0宽度 ('padding' 属性)。第二幅图中效果很明显
- LI盒的margin是透明的——外边距总是透明的——所以，UL的内边距和内容区域的背景色 (yellow) 透出来了
- 第二个LI元素指定了虚线边框 ('border-style' 属性)

8.3 外边距属性： 'margin-top'，'margin-right'，'margin-bottom'，'margin-left'和'margin'

外边距属性指定了盒的外边距区的宽度。'margin' 简写属性一次性设置四周的外边距，而其它外边距属性只设置它们各侧的。这些属性适用于所有元素，但非替换的行内元素上的垂直margin将不会产生任何效果

本节定义的属性涉及<margin-width>值类型，可以取下列值之一：

<length>
指定一个固定宽度

<percentage>

百分比根据生成盒的包含块的width来计算。 注意，这一点对于'margin-top'和'margin-bottom'也适用。 如果包含块的宽度取决于该元素，那么产生的布局在CSS 2.1是未定义的
auto
具体行为见[计算width与margin](#)章节

margin属性允许负值，但可能存在具体实现限制

'margin-top', 'margin-bottom'

| | |
|-----------------|---|
| Value: | <margin-width> inherit |
| Initial: | 0 |
| Applies to: | 除display类型为table系列中除了table-caption，table和inline-table以外的所有元素（译注：也就是说，table系列display值中，margin-*只适用于table-caption, table, inline-table，其余的都不适用，但margin-*对非table系列都适用） |
| Inherited: | no |
| Percentages: | 参照包含块的宽度 |
| Media: | visual |
| Computed value: | 指定的百分比或者绝对长度 |

这两个属性对非替换的行内元素无效

'margin-right', 'margin-left'

| | |
|-----------------|---|
| Value: | <margin-width> inherit |
| Initial: | 0 |
| Applies to: | 除display类型为table系列中除了table-caption，table和inline-table以外的所有元素（译注：也就是说，table系列display值中，margin-*只适用于table-caption, table, inline-table，其余的都不适用，但margin-*对非table系列都适用） |
| Inherited: | no |
| Percentages: | 参照包含块的宽度 |
| Media: | visual |
| Computed value: | 指定的百分比或者绝对长度 |

这些属性给定了盒的上，右，下和左外边距

```
h1 { margin-top: 2em }
```

'margin'

| | |
|-----------------|---|
| Value: | <margin-width> {1,4} inherit |
| Initial: | 见单独的各个属性 |
| Applies to: | 除display类型为table系列中除了table-caption，table和inline-table以外的所有元素（译注：也就是说，table系列display值中，margin-*只适用于table-caption, table, inline-table，其余的都不适用，但margin-*对非table系列都适用） |
| Inherited: | no |
| Percentages: | 参照包含块的宽度 |
| Media: | visual |
| Computed value: | 见单独的各个属性 |

'margin'属性是一个简写属性，用来在样式表中一次性设置'margin-top'，'margin-right'，'margin-bottom'和'margin-left'

如果只有1个分量值，它会应用于所有方向。如果有2个值，top和bottom margin被设置为第一个值，right和left margin被设置为第二个值。如果有3个值，top被设置为第一个值，left和right被设置

为第二个值，**bottom**被设置为第三个值。如果有4个值，会分别应用于**top**，**right**，**bottom**和**left**

```
body { margin: 2em } /* all margins set to 2em */
body { margin: 1em 2em } /* top & bottom = 1em, right & left = 2em */
body { margin: 1em 2em 3em } /* top=1em, right=2em, bottom=3em, left=2em */
```

上例的最后一条规则等价于下例：

```
body {
  margin-top: 1em;
  margin-right: 2em;
  margin-bottom: 3em;
  margin-left: 2em; /* copied from opposite side (right) */
}
```

8.3.1 合并外边距

CSS中，两个或多个盒（可能但不一定是兄弟）的相邻的**margin**会被结合成一个**margin**。外边距按这种方式结合叫做合并（*collapse*），产生的结合的外边距叫做折叠外边距（*collapsed margin* 译注：这里译作折叠表示结果，与合并的动作区分开）

相邻的垂直外边距会合并，除了：

- 根元素盒的**margin**不合并
- 如果一个带有**间隙**（[clearance](#) 译注：是指clear属性导致元素位置移动形成的间隙）的元素的上下外边距与下外边距相邻，它的外边距会和紧挨着的兄弟（元素）的相邻外边距合并，但合并后不会再和父级块的下外边距合并

水平**margin**不会合并

两个**margin**是相邻的，当且仅当：

- 都属于流内（in-flow）[块级盒](#)，处于同一个[块格式化上下文](#)
- 没有行框（line box），空隙，内边距和边框把它们隔开（注意，因此某些0高度行框（见9.4.2）会被忽略）
- 都属于垂直相邻框边界（vertically-adjacent box edges），即形成下列某一对：
 - 盒的上边距与其第一个流内（in-flow）孩子的上边距
 - 盒的下边距与其下一个流内紧挨着的兄弟的上边距
 - 最后一个流内孩子的下边距与其height计算值为'auto'的父元素的下边距
 - 盒的上边距和下边距，要求该盒没有建立新的块格式化上下文，并且'**min-height**'计算值为0，'**height**'计算值为0或'auto'，还没有流内孩子

折叠外边距也能与另一个外边距相邻，只要其外边距的任意一部分与那个外边距相邻就算

注意 相邻外边距也可以由不具兄弟或祖先关系的元素生成

注意 上面的规则表明：

- [浮动的盒](#)与任何其它盒之间的**margin**不会合并（甚至一个浮动盒与它的流内子级之间也不会）
- 建立了新的块格式化上下文的元素（例如，浮动盒与'[overflow](#)'不为'visible'的元素）的**margin**不会与它们的流内孩子合并
- [绝对定位](#)的盒的**margin**不会合并（甚至与它们的流内子级也不会）
- **inline-block**盒的**margin**不会合并（甚至与它们的流内子级也不会）
- 流内块级元素的**bottom margin**总会与它的下一个流内块级兄弟的**top margin**合并，除非该兄弟（元素）具有间隙
- 流内块级元素的**top margin**会与它的第一个流内块级子级的**top margin**合并，条件是该元素没有上边框和上内边距，并且其孩子不具有间隙
- 一个'**height**'为'auto'并且'**min-height**'为0的流内块级盒的**bottom margin**会与它的最后一个流内块级子级的**bottom margin**合并，条件是该盒没有下内边距和下边框，并且其孩子的

下外边距没有与具有间隙的上外边距合并

- 盒自身的外边距也会合并，条件是'min-height'属性为0，既没有上下边框，也没有上下内边距，'height'为0或'auto'，且不含行框的话，那么其所有流内孩子的外边距（如果存在的话）都会合并

当两个或者更多的margin合并时，产生的margin宽度为被合并的外边距宽度中的最大值。至于负margin，就从正相邻margin的最大值中减去负相邻margin的绝对值的最大值。如果没有正margin，就用0减去相邻margin的绝对值的最大值

如果盒的上下外边距相邻，那么外边距合并时可能会穿过它（it is possible for margins to collapse through it）。这种情况下，该元素的位置取决于它与其它外边距被合并了的元素的关系

- 如果该元素的外边距与其父元素的上外边距合并了，盒的上边框边界被定义为与其父元素的相同
- 否则，要么该元素的父元素没参与外边距合并，要么只涉及其父元素的下外边距。该元素上边框边界的位置与元素下边框非0时的位置相同

注意，被折叠外边距穿过的元素的位置不影响其它外边距正要被合并的元素的位置，其上边框边界的位置仅用于布局这些元素的后代元素

8.4 内边距属性：'padding-top'，'padding-right'，'padding-bottom'，'padding-left' 和 'padding'

内边距属性指定了盒的内边距区的宽度。'padding'简写属性一次性设置四周的内边距，而其它内边距属性只设置它们各侧的

本节定义的属性涉及<padding-width>值类型，可以取下列值之一：

<length>

指定一个固定宽度

<percentage>

百分比根据生成盒的包含块的width来计算，即使是'padding-top'和'padding-bottom'。如果包含块的宽度取决于该元素，那么产生的布局在CSS 2.1是未定义的

与margin属性不同，padding值不能为负。其它与margin属性类似，百分比值的padding属性参照其生成盒的包含块的宽度

'padding-top', 'padding-right', 'padding-bottom', 'padding-left'

| | |
|-----------------|--|
| Value: | <padding-width> inherit |
| Initial: | 0 |
| Applies to: | 除table-row-group, table-header-group, table-footer-group, table-row, table-column-group和table-column外的所有元素 |
| Inherited: | no |
| Percentages: | 参照包含块的宽度 |
| Media: | visual |
| Computed value: | 指定的百分比或者绝对长度 |

上面这些属性设置了盒的上，右，下和左内边距

```
blockquote { padding-top: 0.3em }
```

'padding'

| | |
|-------------|--|
| Value: | <padding-width> { 1,4 } inherit |
| Initial: | 见单独的各个属性 |
| Applies to: | 除table-row-group, table-header-group, table-footer-group, table- |

| | |
|-----------------|--|
| | row, table-column-group和table-column外的所有元素 |
| Inherited: | no |
| Percentages: | 参照包含块的宽度 |
| Media: | visual |
| Computed value: | 见单独的各个属性 |

'padding' 属性是个简写属性，用来在样式表中一次性设置'padding-top'，'padding-right'，'padding-bottom'和'padding-left'

如果只有1个分量值，它会应用于所有方向。如果有2个值，top和bottom padding会被设置为第一个值，并且right和left padding会被设置为第二个值。如果有3个值，top被设置为第一个值，left和right被设置为第二个值，bottom被设置为第三个值。如果有4个值，会分别应用于top, right, bottom和left

内边距区的表面颜色或图像通过'background' 属性来指定：

```
h1 {
  background: white;
  padding: 1em 2em;
}
```

上例指定了'1em'的垂直内边距（'padding-top' 和'padding-bottom'）和'2em'的水平内边距（'padding-right'和'padding-left'）。'em'单位与元素的字体大小有关：'1em'等于当前所用字体的大小

8.5 边框属性

边框属性指定了盒的边框区的宽度，颜色和样式。这些属性适用于所有元素

注意 值得注意的是HTML，用户代理可能为某些用户界面元素（例如，按钮，菜单等等）渲染边框，与“普通的”元素不同

8.5.1 边框宽度：'border-top-width'，'border-right-width'，'border-bottom-width'，'border-left-width' 和'border-width'

边框宽度属性指定了边框区的宽度。本节中该属性的定义涉及<border-width>值类型，可以取下列值之一：

- thin 细边框
- medium 中等边框
- thick 粗边框

<length> 边框的粗细有一个明确的值。边框宽度不能为负

前三个值的解释取决于用户代理。但必须保持下列关系：

'thin' <='medium' <='thick'.

而且，这些宽度在一篇文档中必须是一个常量

| | |
|--|--------------------------|
| 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width' | |
| Value: | <border-width> inherit |
| Initial: | medium |
| Applies to: | 所有元素 |
| Inherited: | no |

Percentages: N/A
Media: visual
Computed value: 绝对长度，或者'0'，如果border style为'none'或者'hidden'的话

这些属性设置了盒的上，右，下和左边框的宽度

'border-width'

Value: <border-width> {1,4} | inherit
Initial: 见单独的各个属性
Applies to: 所有元素
Inherited: no
Percentages: N/A
Media: visual
Computed value: 见单独的各个属性

该属性是个简写属性，用来在样式表中一次性设置'border-top-width'，'border-right-width'，'border-bottom-width'和'border-left-width'

如果有1个分量值，会应用于所有方向。如果有2个值，top和bottom border会被设置为第一个值，right 和left会被设置为第二个值。如果有3个值，top会被设置为第一个值，left和right会被设置为第二个值，bottom会被设置为第三个值。如果有4个值，它们会分别应用于top, right, bottom和left

下例中，注释表明了产生的上，右，下和左边框的宽度：

```
h1 { border-width: thin }           /* thin thin thin thin */
h1 { border-width: thin thick }    /* thin thick thin thick */
h1 { border-width: thin thick medium } /* thin thick medium thick */
```

8.5.2 边框颜色：'border-top-color'，'border-right-color'，'border-bottom-color'，'border-left-color' 和'border-color'

边框颜色属性指定了盒边框的颜色

'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color'

Value: <color> | transparent | inherit
Initial: 'color'属性的值
Applies to: 所有元素
Inherited: no
Percentages: N/A
Media: visual
Computed value: 从'color'属性取值的话，取'color'的计算值，否则，就按指定值

'border-color'

Value: [<color> | transparent] {1,4} | inherit
Initial: 见单独的各个属性
Applies to: 所有元素
Inherited: no
Percentages: N/A
Media: visual
Computed value: 见单独的各个属性

'border-color' 一次性设置4条边框的颜色。值含义如下：

<color>
指定一个颜色值

transparent

边框是透明的（尽管它可能具有宽度）

'border-color' 属性可以有1到4个分量值，这些值用来设置不同方向的（边框颜色），与'border-width'类似

如果一个元素没有用border属性指定border color，用户代理必须把该元素的'color'属性的值作为border color的计算值

本例中，border将是黑色实线

```
p {
  color: black;
  background: white;
  border: solid;
}
```

8.5.3 边框样式： 'border-top-style'，'border-right-style'，'border-bottom-style'，'border-left-style' 和'border-style'

边框样式属性指定了盒边框的线条样式（实线，双实线，虚线等等）。本节中该属性的定义涉及<**border-style**>值类型，可以取下列值之一：

- none**
无边框，边框宽度的计算值为0
- hidden**
与'none'相同，除了在表格元素的边框冲突处理时
- dotted**
边框是一系列点
- dashed**
边框是一系列短线段
- solid**
边框是一条线段
- double**
边框是两条实线。两条实线及它们之间的空隙之和等于'border-width'的值
- groove**
边框看起来像是被刻进了画布（译注：3D凹槽效果）
- ridge**
与'groove'相反：边框看起来像是要从画布里跑出来（译注：3D凸槽效果）
- inset**
边框让盒看起来像是被嵌进画布了（译注：没有3D效果）
- outset**
与'inset'相反：边框让盒看起来像是要从画布里出来了（译注：没有3D效果）

所有边框都是在盒背景之上绘制的。border值为'groove', 'ridge', 'inset'和'outset'的边框颜色取决于元素的border color属性，但UA可能会选用它们自己的算法来计算要应用的实际颜色。例如，如果'border-color'值为'silver'，那么一个UA可能用一个从白到深灰的渐变色来表示有斜度的边框（sloping border 译注：应该是指用渐变模拟3D效果的边框）

'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style'

| | |
|-----------------|--------------------------|
| Value: | <border-style> inherit |
| Initial: | none |
| Applies to: | 所有元素 |
| Inherited: | no |
| Percentages: | N/A |
| Media: | visual |
| Computed value: | 与指定值相同 |

'border-style'

- Value: <border-style> { 1,4 } | inherit
- Initial: 见单独的各个属性
- Applies to: 所有元素
- Inherited: no
- Percentages: N/A
- Media: visual
- Computed value: 见单独的各个属性

'border-style' 属性设置了4条边框的样式。它可以有1到4个分量值，这些值用来设置不同方向的（边框样式），与上面的'border-width'类似

```
#xy34 { border-style: solid dotted }
```

上例中，水平边框将是'solid'，垂直边框将是'dotted'

因为border style的初始值为'none'，不会有可见边框，除非设置了border style

8.5.4 边框简写属性: 'border-top', 'border-right', 'border-bottom', 'border-left'和'border'

'border-top', 'border-right', 'border-bottom', 'border-left'

- Value: [<border-width> || <border-style> || <'border-top-color'>_] | inherit
- Initial: 见单独的各个属性
- Applies to: 所有元素
- Inherited: no
- Percentages: N/A
- Media: visual
- Computed value: 见单独的各个属性

这是个简写属性，用来设置盒的上，右，下和左边框的宽度，样式与颜色

```
h1 { border-bottom: thick solid red }
```

上面的规则将设置H1元素下方的边框宽度，样式和颜色。省略的值会被设置为其初始值。因为下面的规则没有指定border color，边框将获得'color'属性指定的颜色：

```
H1 { border-bottom: thick solid }
```

'border'

- Value: [<border-width> || <border-style> || <'border-top-color'>_] | inherit
- Initial: 见单独的各个属性
- Applies to: 所有元素
- Inherited: no
- Percentages: N/A
- Media: visual
- Computed value: 见单独的各个属性

'border' 属性是个简写属性，用来给盒的4条边框设置相同的宽度，颜色和样式。与简写属性'margin'和'padding'不同，'border'属性不能对4条边框设置不同的值。为此，必须使用一个或多个其它border属性

例如，下面第一条规则等价于它后面的4条规则集：

```
p { border: solid red }
p {
  border-top: solid red;
  border-right: solid red;
```

```
border-bottom: solid red;  
border-left: solid red  
}
```

因为，在某种程度上，属性功能有所重叠，（样式）规则的声明顺序很重要

考虑本例：

```
blockquote {  
border: solid red;  
border-left: double;  
color: black;  
}
```

上例中，左边框的颜色是黑色，而其它边框是红色。这是因为'border-left'同时设置了宽度，样式和颜色。因为color值没有通过'border-left'属性给出，它将从'color'属性取。'color'属性确实是在'border-left'属性之后设置的，但没关系

8.6 双向环境（bidirectional context）中行内元素的盒模型

对于每一个行框（line box），UA必须为每个元素生成行内盒（inline box）并按照可视顺序（不是逻辑顺序）呈现外边距，边框和内边距

当元素的'direction'属性为'ltr'时，元素出现的第一个行框最左端生成的盒具有左外边距，左边框和左内边距，并且元素出现的最后一个行框最右端生成的盒具有右内边距，右边框和右外边距

当元素的'direction'属性为'rtl'时，元素出现的第一个行框最右端生成的盒具有右内边距，右边框和右外边距，并且元素出现的最后一个行框最左端生成的盒具有左外边距，左边框和左内边距

[上一章](#) [下一章](#) [内容](#) [属性](#) [索引](#)